In [1]:
```
from DeepCompressor import *
ce = capacity_estimator()
```

In [2]:
```
vgg16 = model('vgg16', True)
vgg16.set_capacity_estimator(ce)
vgg16.parallelize()
vgg16.prepare_dataset()
```

In [3]:
```
num_epochs = 3
max_epochs_stop = 3
num_classes = 10
batch_size = 10
learning_rate = 0.001
print_every = 1
```

In [4]:
```
training_params = [num_epochs, max_epochs_stop, num_classes, batch_size, learn
ing_rate, print_every]
deep_compressor = deep_compressor(vgg16, trainer(vgg16, training_params))
```

In [5]:

```
deep_compressor.train(dropout=100)
deep_compressor.test() # ce.estimate(25088, 4096, 4096, 256, 10)
deep_compressor.squeeze(100)
deep_compressor.train(dropout=100)
deep_compressor.test() # ce.estimate(25088, 4096-100, 4096, 256, 10)
deep_compressor.squeeze(100)
deep_compressor.train(dropout=100)
deep_compressor.test() # ce.estimate(25088, 4096-100, 4096, 256, 10)
deep_compressor.squeeze(100)
deep_compressor.train(dropout=100)
deep_compressor.test() # ce.estimate(25088, 4096-100, 4096, 256, 10)
deep_compressor.squeeze(100)
deep_compressor.train(dropout=100)
deep_compressor.test() # ce.estimate(25088, 4096-100, 4096, 256, 10)
deep_compressor.squeeze(100)
deep_compressor.train(dropout=100)
deep_compressor.test() # ce.estimate(25088, 4096-100, 4096, 256, 10)
```

```
Starting Training from Scratch.

Epoch: 0        100.00% complete. 116.78 seconds elapsed in epoch.
Epoch: 0        Training Loss: 0.8359   Validation Loss: 3.3102
                Training Accuracy: 71.79%       Validation Accuracy: 17.68%
Epoch: 1        100.00% complete. 116.15 seconds elapsed in epoch.
Epoch: 1        Training Loss: 0.4978   Validation Loss: 3.7030
                Training Accuracy: 83.82%       Validation Accuracy: 17.06%
Epoch: 2        100.00% complete. 115.98 seconds elapsed in epoch.
Epoch: 2        Training Loss: 0.4283   Validation Loss: 3.5333
                Training Accuracy: 86.17%       Validation Accuracy: 19.37%

Test Loss: 0.2600673564337194   Test Accuracy: 0.9200
Starting Training from Scratch.

Epoch: 0        100.00% complete. 115.56 seconds elapsed in epoch.
Epoch: 0        Training Loss: 0.3937   Validation Loss: 3.6632
                Training Accuracy: 87.40%       Validation Accuracy: 17.96%
Epoch: 1        100.00% complete. 115.80 seconds elapsed in epoch.
Epoch: 1        Training Loss: 0.3634   Validation Loss: 3.6936
                Training Accuracy: 88.30%       Validation Accuracy: 19.72%
Epoch: 2        100.00% complete. 116.20 seconds elapsed in epoch.
Epoch: 2        Training Loss: 0.3460   Validation Loss: 3.2967
                Training Accuracy: 89.08%       Validation Accuracy: 23.57%

Test Loss: 0.2043949328083545   Test Accuracy: 0.9349
Starting Training from Scratch.

Epoch: 0        100.00% complete. 116.34 seconds elapsed in epoch.
Epoch: 0        Training Loss: 0.3275   Validation Loss: 3.6264
                Training Accuracy: 89.60%       Validation Accuracy: 22.23%
Epoch: 1        100.00% complete. 116.43 seconds elapsed in epoch.
Epoch: 1        Training Loss: 0.3153   Validation Loss: 4.3907
                Training Accuracy: 89.93%       Validation Accuracy: 21.11%
Epoch: 2        100.00% complete. 116.18 seconds elapsed in epoch.
Epoch: 2        Training Loss: 0.2990   Validation Loss: 4.5824
                Training Accuracy: 90.59%       Validation Accuracy: 21.06%

Test Loss: 0.18121262782253325  Test Accuracy: 0.9458
Starting Training from Scratch.

Epoch: 0        100.00% complete. 116.94 seconds elapsed in epoch.
Epoch: 0        Training Loss: 0.2963   Validation Loss: 4.9970
                Training Accuracy: 90.76%       Validation Accuracy: 20.72%
Epoch: 1        100.00% complete. 115.82 seconds elapsed in epoch.
Epoch: 1        Training Loss: 0.2806   Validation Loss: 5.1167
                Training Accuracy: 91.14%       Validation Accuracy: 18.91%
Epoch: 2        100.00% complete. 116.25 seconds elapsed in epoch.
Epoch: 2        Training Loss: 0.2813   Validation Loss: 5.8718
                Training Accuracy: 91.08%       Validation Accuracy: 17.24%

Test Loss: 0.1778415240859613   Test Accuracy: 0.9450
Starting Training from Scratch.

Epoch: 0        100.00% complete. 115.75 seconds elapsed in epoch.
Epoch: 0        Training Loss: 0.2704   Validation Loss: 6.2654
                Training Accuracy: 91.51%       Validation Accuracy: 19.70%
```

```
Epoch: 1        100.00% complete. 115.84 seconds elapsed in epoch.
Epoch: 1        Training Loss: 0.2650   Validation Loss: 5.2456
                Training Accuracy: 91.55%       Validation Accuracy: 19.04%
Epoch: 2        100.00% complete. 115.65 seconds elapsed in epoch.
Epoch: 2        Training Loss: 0.2623   Validation Loss: 6.3384
                Training Accuracy: 91.74%       Validation Accuracy: 18.16%

Test Loss: 0.1658061691885814   Test Accuracy: 0.9461
Starting Training from Scratch.

Epoch: 0        100.00% complete. 116.83 seconds elapsed in epoch.
Epoch: 0        Training Loss: 0.2625   Validation Loss: 5.6887
                Training Accuracy: 91.80%       Validation Accuracy: 19.52%
Epoch: 1        100.00% complete. 117.13 seconds elapsed in epoch.
Epoch: 1        Training Loss: 0.2559   Validation Loss: 6.7967
                Training Accuracy: 91.95%       Validation Accuracy: 17.63%
Epoch: 2        100.00% complete. 116.48 seconds elapsed in epoch.
Epoch: 2        Training Loss: 0.2493   Validation Loss: 6.7890
                Training Accuracy: 92.20%       Validation Accuracy: 17.33%

Test Loss: 0.16473693074542098  Test Accuracy: 0.9493
```

In [6]:
```
df = pd.DataFrame(data=deep_compressor.model.history)
df
```

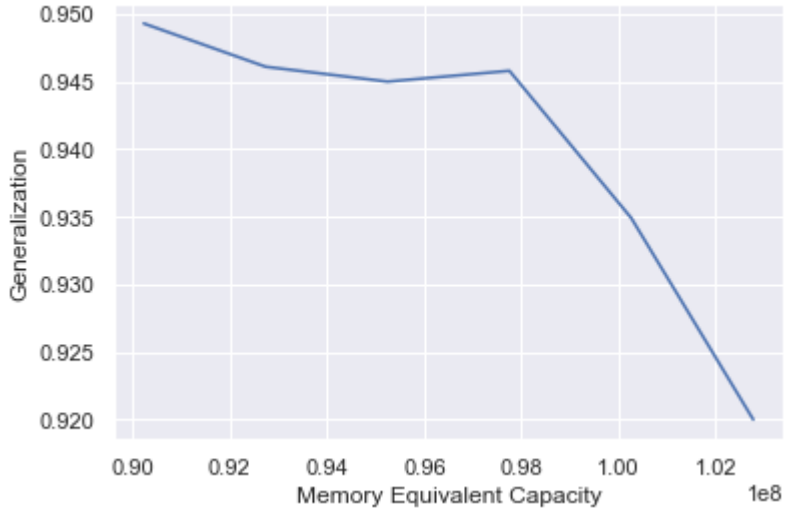
Out[6]:

	acc	cap
0	0.9200	102772992
1	0.9349	100263992
2	0.9458	97754992
3	0.9450	95245992
4	0.9461	92736992
5	0.9493	90227992

In [9]:
```
import seaborn as sns
sns.set(style="darkgrid")

# Plot the responses for different events and regions
fig = sns.lineplot(x='cap', y='acc',
             data=df);
fig.set(ylabel='Generalization', xlabel='Memory Equivalent Capacity')
```

Out[9]:
```
[Text(0, 0.5, 'Generalization'), Text(0.5, 0, 'Memory Equivalent Capacity')]
```



In [10]:
```
fig.get_figure().savefig('GC_curve.jpg')
```